UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/083,872	02/27/2002	Eric DeLano	10016663-1	4721

7590 04/17/2007

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400

EXAMINER	
LI, AIMEE J	
ART UNIT	**PAPER NUMBER**
2183	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	04/17/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No. 10/083,872	Applicant(s) DELANO, ERIC	
	Examiner Aimee J. Li	Art Unit 2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 January 2007*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *19-34* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *19-34* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *04 April 2005* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. Claims 19-34 have been considered. Claims 1-10, 12, 13, 15, 16, and 18 have been cancelled as per Applicants' request. New claims 19-34 have been added as per Applicants' request.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: RCE as filed 31 January 2007 and Amendment as filed 31 January 2007.

Continued Examination Under 37 CFR 1.114

3. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 31 January 2007 has been entered.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 19-34 are rejected under 35 U.S.C. 102(e) as being taught by Fernando et al., U.S. Patent Number 6,272,616 (herein referred to as Fernando).

6. Referring to claim 19, Fernando has taught a method for processing bundled instructions through execution units of a processor, comprising the steps of:

a. Determining a mode of operation, wherein the mode of operation comprises one of a throughput mode and a wide mode (Fernando column 2, lines 51-62 "...The architecture is capable of running in various modes, including single threaded mode, SIMD mode and MIM mode..."; column 3, line 29-33 "...at least three different modes of parallelism..."; column 6, lines 33-40 "...CFORK instruction is one of the special instructions and, specifically, is the instruction which activates one or more secondary instruction pipelines in the SIMD mode..."; column 7, lines 14-35 "...DFORK is one of the special instructions. This instruction indicates that the architecture is to enter the MIMD mode..."; and Figure 5);

b. In the throughput mode:

i. Fetching a first bundle of instructions from a first thread of a multiply threaded program (Fernando column 3, line 62-67 "In the MIMD mode, the full hardware of two or more of the instruction pipelines are fully operations...each instruction pipeline individually fetches instructions from the instruction memory using its own fetch stage and fetches its own data..."; column 7, lines 6-46 "...The two threads **16** and **18** then will operate essentially independently of each other, reading and executing different instructions and using different data..."; column 11, line 37 to column 12, line 4 "...a DFORK instruction causing instruction pipeline **2** to being operating in MIMD mode at time t**2**…"; Figure 4; Figure 5; and Figure 7);

ii. Distributing the first bundle to a first cluster of the execution units for execution therethrough (Fernando column 3, line 62-67 "In the MIMD mode, the full hardware of two or more of the instruction pipelines are fully operations…each instruction pipeline individually fetches instructions from the instruction memory using its own fetch stage and fetches its own data…"; column 7, lines 6-46 "…The two threads **16** and **18** then will operate essentially independently of each other, reading and executing different instructions and using different data…"; column 11, line 37 to column 12, line 4 "…a DFORK instruction causing instruction pipeline **2** to being operating in MIMD mode at time t**2**…"; Figure 4; Figure 5; and Figure 7);

iii. Fetching a second bundle of instructions from a second thread of the program (Fernando column 3, line 62-67 "In the MIMD mode, the full hardware of two or more of the instruction pipelines are fully operations…each instruction pipeline individually fetches instructions from the instruction memory using its own fetch stage and fetches its own data…"; column 7, lines 6-46 "…The two threads **16** and **18** then will operate essentially independently of each other, reading and executing different instructions and using different data…"; column 11, line 37 to column 12, line 4 "…a DFORK instruction causing instruction pipeline **2** to being operating in MIMD mode at time **t2**…"; Figure 4; Figure 5; and Figure 7 – In regards to Fernando, in MIMD mode, each instruction pipeline works independent of the others to fetch instructions from the individual threads associated with the instruction pipeline and to execute those instructions.); and

iv. Distributing the second bundle to a second cluster of the execution units for execution therethrough (Fernando column 3, line 62-67 "In the MIMD mode, the full hardware of two or more of the instruction pipelines are fully operations…each instruction pipeline individually fetches instructions from the instruction memory using its own fetch stage and fetches its own data…"; column 7, lines 6-46 "…The two threads **16** and **18** then will operate essentially independently of each other, reading and executing different instructions and using different data…"; column 11, line 37 to column 12, line 4 "…a DFORK instruction causing instruction pipeline **2** to being operating in MIMD mode at time **t2**…"; Figure 4; Figure 5; and Figure 7); and

c. In the wide mode:

i. Fetching a third bundle of instructions from a third thread of the program (Fernando column 3, lines 47-61 "In the SIMD mode, the fetch stage, decoder stage, and execute stage of the first of the parallel instruction pipelines is fully operational. One or more of the decoder stages of the other instruction pipelines are coupled to received the instructions fetched by the fetch stage of the first instruction pipeline…"; column 6, lines 33-54 "…Secondary instruction pipeline **18** then commences to execute the code fetched by fetch stage **20**a, while primary instruction pipeline **16** continues to execute the same instructions."; column 11, line 37 to column 12, line 4 "…At time t**9**, a CFORK instruction is executed in instruction pipeline **1** activating instruction pipeline **2** in the SIMD mode…"; Figure 3; Figure 5; and Figure 7);

ii. Distributing the third bundle to the first cluster for execution therethrough (Fernando column 3, lines 47-61 "In the SIMD mode, the fetch stage, decoder stage, and execute stage of the first of the parallel instruction pipelines is fully operational. One or more of the decoder stages of the other instruction pipelines are coupled to received the instructions fetched by the fetch stage of the first instruction pipeline…"; column 6, lines 33-54 "…Secondary instruction pipeline **18** then commences to execute the code fetched by fetch stage **20**a, while primary instruction pipeline **16** continues to execute the same instructions."; column 11, line 37 to column 12, line 4 "…At time t**9**, a CFORK instruction is executed in instruction pipeline **1** activating instruction pipeline **2** in the SIMD mode…"; Figure 3; Figure 5; and Figure 7);

iii. Fetching a fourth bundle of instructions from the third thread of the program (Fernando column 3, lines 47-61 "In the SIMD mode, the fetch stage, decoder stage, and execute stage of the first of the parallel instruction pipelines is fully operational. One or more of the decoder stages of the other instruction pipelines are coupled to received the instructions fetched by the fetch stage of the first instruction pipeline…"; column 6, lines 33-54 "…Secondary instruction pipeline **18** then commences to execute the code fetched by fetch stage **20**a, while primary instruction pipeline **16** continues to execute the same instructions."; column 11, line 37 to column 12, line 4 "…At time t**9**, a CFORK instruction is executed in instruction pipeline **1** activating instruction pipeline **2** in the SIMD mode…"; Figure 3; Figure 5; and Figure 7 – In regards to Fernando, in SIMD mode, the two instruction pipelines execute instructions from the same thread, so all bundles fetched from this thread, including the third and fourth bundles, are executed in both the first and second instruction pipelines. Also, there is nothing in the claim language stating that the fourth bundle of instructions is different from the third bundle of instructions. In addition, Fernando has taught in column 1, line 54 to column 2, line 3 the use of VLIW instructions where each VLIW instruction fetch contains multiple single operation instructions for parallel execution in a datapath such as those shown in Figure 1.); and

iv. Distributing the fourth bundle to the second cluster for execution therethrough (Fernando column 3, lines 47-61 "In the SIMD mode, the fetch stage, decoder stage, and execute stage of the first of the parallel instruction pipelines is fully operational. One or more of the decoder stages of the other instruction pipelines are coupled to received the instructions fetched by the fetch stage of the first instruction pipeline…"; column 6, lines 33-54 "…Secondary instruction pipeline **18** then commences to execute the code fetched by fetch stage **20**a, while primary instruction pipeline **16** continues to execute the same instructions."; column 11, line 37 to column 12, line 4 "…At time t**9**, a CFORK instruction is executed in instruction pipeline **1** activating instruction pipeline **2** in the SIMD mode…"; Figure 3; Figure 5; and Figure 7).

7. Referring to claim 20, Fernando has taught the method of claim 19, further comprising:

   a. Processing the first and third bundles within the first cluster (Fernando column 11, line 37 to column 12, line 4 "…instruction pipelines **1**, **2** and **3** operate independently of each other in MIMD mode…instruction pipeline **2** starts to accept instructions from the fetch stage in instruction pipeline **1**…" and Figure 7);

   b. Processing the second and fourth bundles within the second cluster (Fernando column 11, line 37 to column 12, line 4 "…instruction pipelines **1**, **2** and **3** operate independently of each other in MIMD mode…instruction pipeline **2** starts to accept instructions from the fetch stage in instruction pipeline **1**…" and Figure 7).

8. Referring to claim 21, Fernando has taught the method of claim 19, further comprising the step of architecting data from the first cluster to a first register file (Fernando column 4, line 57 to column 5, line 3 "…There is a register **25** for each of the execute stages…for temporarily storing data…data to be operated on by the instructions. The register files of all of the pipelines…" and Figure 1).

9. Referring to claim 22, Fernando has taught the method of claim 19, further comprising the step of architecting data from the second cluster to a second register file (Fernando column 4, line 57 to column 5, line 3 "...There is a register **25** for each of the execute stages...for temporarily storing data...data to be operated on by the instructions. The register files of all of the pipelines..." and Figure 1).

10. Referring to claim 23, Fernando has taught The method of claim 19, the step of fetching the first bundle comprising decoding instructions into the first bundle (Fernando column 1, line 54 to column 2, line 3 "...code is either written or compiled to cause such independent instructions to be grouped into a VLIW. Each VLIW is parsed and then fed into multiple issue slots in the processor..."; column 4, lines 46-49 "Decoder stages..."; and Figure 1).

11. Referring to claim 24, Fernando has taught the method of claim 19, the step of fetching the second bundle comprising decoding instructions into the second bundle (Fernando column 1, line 54 to column 2, line 3 "...code is either written or compiled to cause such independent instructions to be grouped into a VLIW. Each VLIW is parsed and then fed into multiple issue slots in the processor..."; column 4, lines 46-49 "Decoder stages..."; and Figure 1).

12. Referring to claim 25, Fernando has taught the method of claim 19, the step of fetching the third bundle comprising decoding instructions into the third bundle (Fernando column 1, line 54 to column 2, line 3 "...code is either written or compiled to cause such independent instructions to be grouped into a VLIW. Each VLIW is parsed and then fed into multiple issue slots in the processor..."; column 4, lines 46-49 "Decoder stages..."; and Figure 1).

13. Referring to claim 26, Fernando has taught the method of claim 19, the step of fetching the fourth bundle comprising decoding instructions into the fourth bundle (Fernando column 1,

line 54 to column 2, line 3 "...code is either written or compiled to cause such independent instructions to be grouped into a VLIW. Each VLIW is parsed and then fed into multiple issue slots in the processor..."; column 4, lines 46-49 "Decoder stages..."; and Figure 1).

14. Referring to claim 27, Fernando has taught the method of claim 19, further comprising bypassing data between the first cluster and the second cluster, as needed, to facilitate the processing of the third bundle through the first cluster and the fourth bundle through the second cluster (Fernando column 4, line 64 to column 5, line 3 "...data may be exchanged) responsive to the appropriate instruction or instructions) between the register file **25*a*** of execute stage **24*a*** of the primary instruction pipeline **16** and the register files **25*b*** of the execute stage **24*b*** of the secondary instruction pipeline **18** via a register bus **38**..." and Figure 1, element 38).

15. Referring to claim 28, Fernando has taught the method of claim 27, wherein the step of bypassing the data utilizes a latch to couple the data between the first cluster and the second cluster (Fernando column 4, line 64 to column 5, line 3 "...data may be exchanged) responsive to the appropriate instruction or instructions) between the register file **25*a*** of execute stage **24*a*** of the primary instruction pipeline **16** and the register files **25*b*** of the execute stage **24*b*** of the secondary instruction pipeline **18** via a register bus **38**..." and Figure 1, element 38 – In regards to Fernando, registers are comprised of latches that store data, so the registers which the data is transferred from and the data is transferred to are latches that a used when passing data to one instruction pipeline from another instruction pipeline, i.e. between the two pipelines. For more information about registers and latches, please see the accompanying excerpt from Heuring and Jordan's <u>Computer Systems Design and Architecture</u> page 151, Register Implementation Domain "In designing the registers, the main choice is between edge-triggered and level-

sensitive flip-flops...Level-sensitive flip-flops, often called *latches*..." and page 534-537, paragraph 3 "...We refer to such an arrangement of flip-flops as a *register*...").

16. Referring to claim 29, Fernando has taught the method of claim 19, wherein the step of determining the mode of operation comprises determining a state of a configuration bit (Fernando column 2, lines 51-62 "...The architecture is capable of running in various modes, including single threaded mode, SIMD mode and MIM mode..."; column 3, line 29-33 "...at least three different modes of parallelism..."; column 6, lines 33-40 "...CFORK instruction is one of the special instructions and, specifically, is the instruction which activates one or more secondary instruction pipelines in the SIMD mode..."; column 7, lines 14-35 "...DFORK is one of the special instructions. This instruction indicates that the architecture is to enter the MIMD mode..."; and Figure 5 – In regards to Fernando, the instructions are comprised of bits that signal when the system is in SIMD or MIMD modes.).

17. Referring to claim 30, Fernando has taught The method of claim 19, further comprising the steps of:

a. In the throughput mode:

i. Fetching a fifth bundle of instructions from a fourth thread of the program (Fernando column 3, line 62-67 "In the MIMD mode, the full hardware of two or more of the instruction pipelines are fully operations...each instruction pipeline individually fetches instructions from the instruction memory using its own fetch stage and fetches its own data..."; column 7, lines 6-46 "...The two threads **16** and **18** then will operate essentially independently of each other, reading and executing different instructions and using different data…"; column 11, line 37 to column 12, line 4 "…a DFORK instruction causing instruction pipeline **2** to being operating in MIMD mode at time t**2**…"; Figure 4; Figure 5; and Figure 7 – In regards to Fernando, Fernando teaches in column 5, line 54 to column 6, line 3 that portions of the program benefit from the use of MIMD and SIMD modes to improve processing performance and speed, meaning that using the DFORK instruction would improve performance in certain portions of the code, so using the DFORK instruction in all possible instances to fetch instructions from new threads is taught.);

ii. Distributing the fifth bundle to the first cluster for execution therethrough (Fernando column 3, line 62-67 "In the MIMD mode, the full hardware of two or more of the instruction pipelines are fully operations…each instruction pipeline individually fetches instructions from the instruction memory using its own fetch stage and fetches its own data…"; column 7, lines 6-46 "…The two threads **16** and **18** then will operate essentially independently of each other, reading and executing different instructions and using different data…"; column 11, line 37 to column 12, line 4 "…a DFORK instruction causing instruction pipeline **2** to being operating in MIMD mode at time t**2**…"; Figure 4; Figure 5; and Figure 7);

iii. Fetching a sixth bundle of instructions from a fifth thread of the program (Fernando column 3, line 62-67 "In the MIMD mode, the full hardware of two or more of the instruction pipelines are fully operations…each instruction pipeline individually fetches instructions from the instruction memory using its own fetch stage and fetches its own data…"; column 7, lines 6-46 "…The two threads **16** and **18** then will operate essentially independently of each other, reading and executing different instructions and using different data…"; column 11, line 37 to column 12, line 4 "…a DFORK instruction causing instruction pipeline **2** to being operating in MIMD mode at time t**2**…"; Figure 4; Figure 5; and Figure 7); and

iv. Distributing the sixth bundle to the second cluster for execution therethrough (Fernando column 3, line 62-67 "In the MIMD mode, the full hardware of two or more of the instruction pipelines are fully operations…each instruction pipeline individually fetches instructions from the instruction memory using its own fetch stage and fetches its own data…"; column 7, lines 6-46 "…The two threads **16** and **18** then will operate essentially independently of each other, reading and executing different instructions and using different data…"; column 11, line 37 to column 12, line 4 "…a DFORK instruction causing instruction pipeline **2** to being operating in MIMD mode at time t**2**…"; Figure 4; Figure 5; and Figure 7).

18. Referring to claim 31, claim 31 contains similar limitations to those found in claims 19 and 29 above and is rejected for similar reasons. The only difference between claim 31 and claims 19 and 29 is that claim 31 is a processor claim while claims 19 and 29 are method claims.

19. Referring to claim 32, claim 32 contains similar limitations to claims 21 and 22 above and is rejected for similar reasons. The only difference between claim 32 and claims 21 and 22 is that claim 32 is a processor claim while claims 21 and 22 are method claims.

20. Referring to claim 33, claim 33 contains similar limitations to claims 27 and 28 above and is rejected for similar reasons. The only difference between claim 33 and claims 27 and 28 are that claim 33 is a processor claim while claims 27 and 28 are method claims.

21. Referring to claim 34, claim 34 contains similar limitations to claims 20 and 30 above and is rejected for similar reasons. The only difference between claim 34 and claims 20 and 30 is that claim 34 is a processor claim while claims 20 and 30 are method claims.

Response to Arguments

22. Applicant's arguments with respect to claims 19-24 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

23. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

a. Pechanek et al., U.S. Patent Numbers 6,151,668; 6,446,191; and PCT WO 99/24903, have taught a VLIW instruction, i.e. instruction bundles, execution system with SIMD, i.e. wide, mode and MIMD, i.e. throughput, mode controlled by instructions designating SIMD or MIMD mode operation.

b. Shoemaker et al., U.S. Patent Application Publication 2003/0135711, has taught a multi-threaded system that executes multiple threads simultaneously when the processor is wide enough, i.e. wide mode, or executes the threads individually when the processor is not wide enough, i.e. throughput mode.

c. Kogge, U.S. Patent Number 5,475,856, has taught a system with single instruction single data mode; SIMD, i.e. wide, mode; and MIMD, i.e. throughput, mode with a switch controlling the mode.

d. MacMillan, U.S. Patent Number 5,689,677, has taught a system with a SIMD, i.e. wide, mode of operation.

e. Sgro et al., U.S. Patent Number 5,903,711, has taught a system with selecting means to control SIMD, i.e. wide, mode and MIMD, i.e. throughput, mode of operation.

f. Gosior et al., U.S. Patent Application Publication 2003/0093655, has taught a system with multi-threaded processes that has SIMD, i.e. wide, and MIMD, i.e. throughput, modes of operation.

24. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aimee J. Li whose telephone number is (571) 272-4169. The examiner can normally be reached on M-T 7:00am-4:30pm.

25. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

26. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Aimee J. Li

Aimee J. Li
25 March 2007